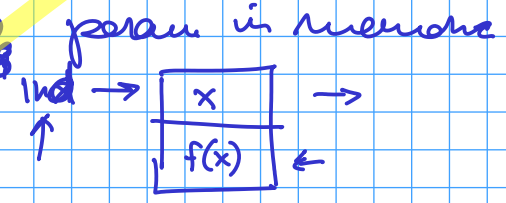
```
int f(int x) { int z = a x^2 + b x + c; return x; }
R10   R9
for (i=0; i<10; i++)
    x[i] = f(x[i]);
```

1 input
1 output

Pass. dei parametri
- registri
- memoria

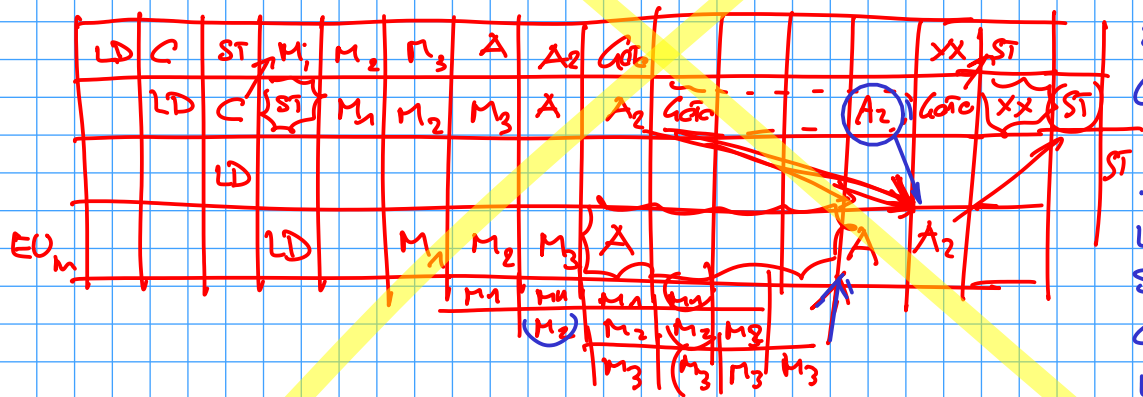

```
loop: LOAD  Rx, Ri, R9
      CALL  Rf, Rret
      STORE Rx, Ri, R10
```

```
f: MUL  R9 R9 R2
   MUL  R2 Ra R3
   MUL  R9 Rb R4
   ADD  R3 R4 R5
   ADD  R5 R6 R10
   GOTO Rret
   XX
```

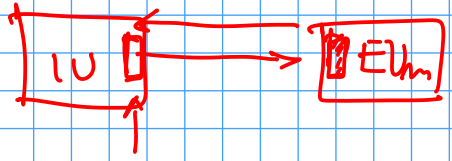


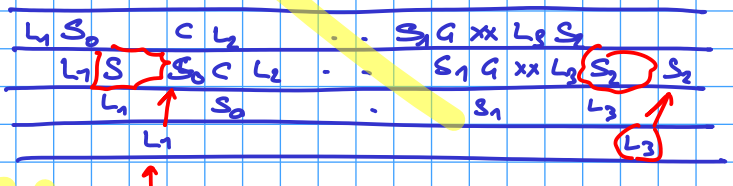

```
f: LOAD2  R8, R0, R9
   {...
   STORE1 R8, #1, R10
   GOTO   Rret
   ...}
```

Passaggio di parametri x registro

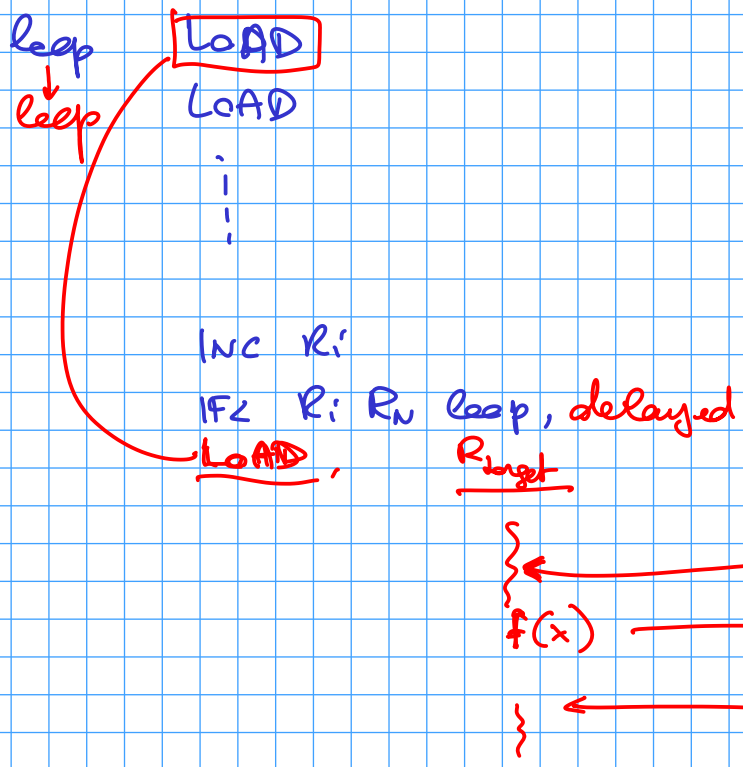


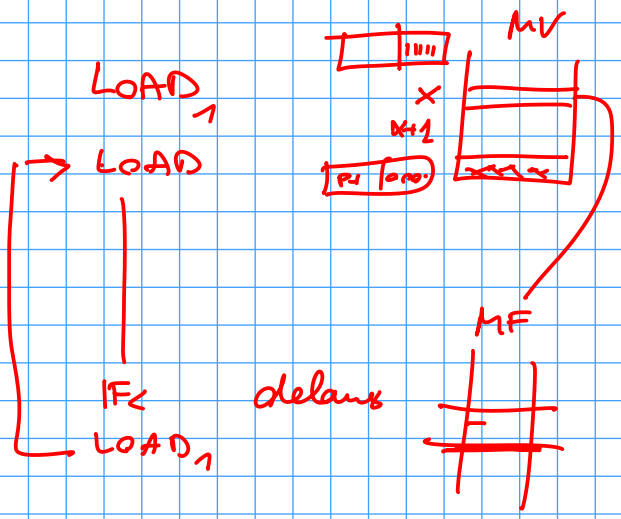

```
LOAD1  Rx, Ri, R1
STOREo R8, 0, R1
CALL   Rf, Rret
LOAD3  R8, #1, R1
STORE  Rx, Ri, R1
```



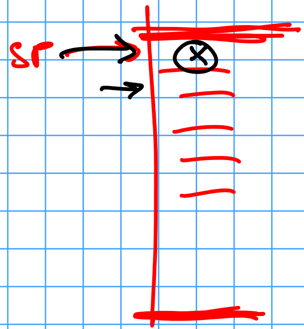


Attenzione: in questa rappresentazione non sono dimenticato di considerare lo slip $MUL_1 \rightarrow MUL_2$ (VEDI ULTIMA PAG. delle NOTE)


loop
LOAD
loop
LOAD
INC Ri
IFZ Ri RN loop, delayed
LOAD.


LOAD1
LOAD
IFZ
LOAD1
delays
MV
MF

Rtarget
f(x)



R_{sp}

int f(int x) { ... }

f: DEC R_{sp}
LOAD R_{sp}, R_0, R_9 } LOAD R_{sp} #-1, [illegible]
⋮
STORE R_{sp}, R_9, R_{10} } STORE R_{sp} #-1, R_{10}
INC R_{sp}
GOTO R_{ret}

però
Non è
[illegible]
ricorsivo

Lh1

```
for(i=0; i<N; i++)
    x[i] = f(x[i]);
```

Lh2

```
sum=0;
for(i=0; i<N; i++)
    sum += x[i];
avg = sum/N;
```

$f(x) = ax^2 + bx + c$

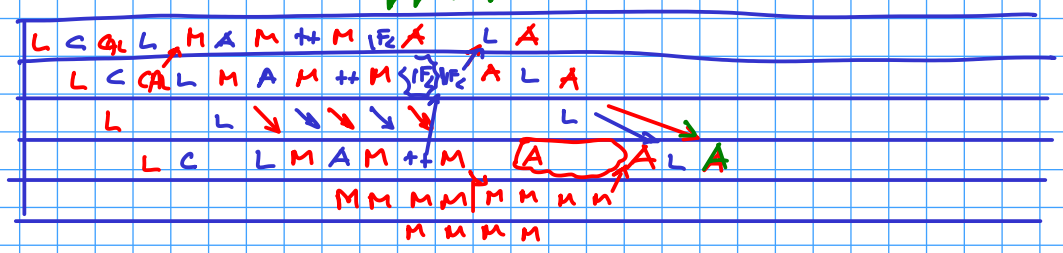
```
loop: LOAD Rx, Ri, Rq
      CALL Rf, Rret
      STORE Rx, Ri, R10
      END
```

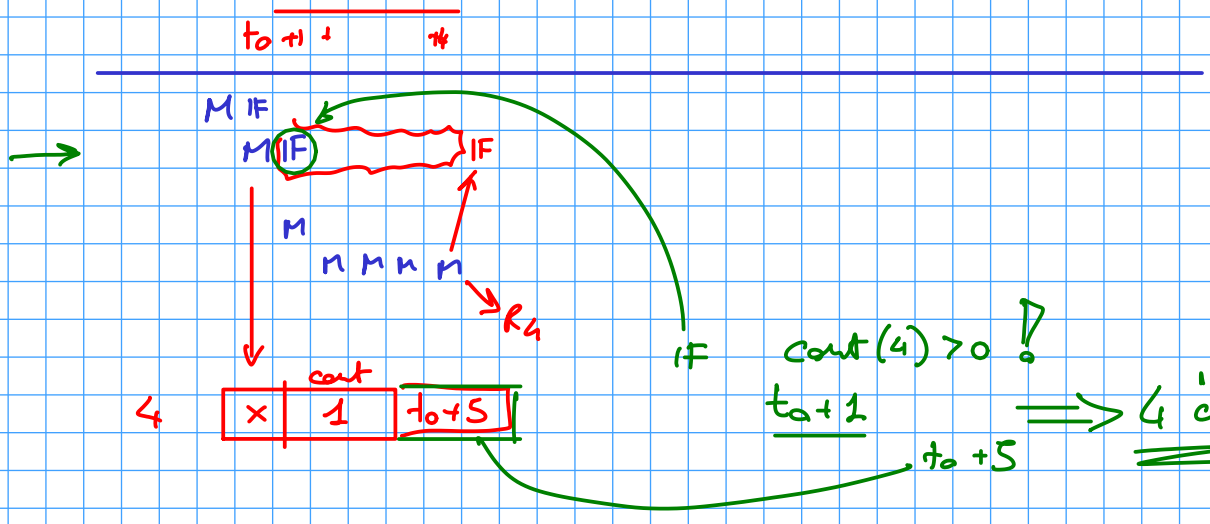
```
f: MUL Rq Rq R2
   MUL R2 Ra R3
   MUL Rq Rb R4
   ADD R3 R4 R5
   ADD R5, Rc, R1c
   GOTO Rret
   xx
```

```
      CLEAR Rsum
loop: LOAD Rx, Ri, R1
      ADD Rsum, R1, Rsum
      INC Ri
      IF2 Ri, RN, loop
cont: DIV Rsum, RN, Ravg
      END
```

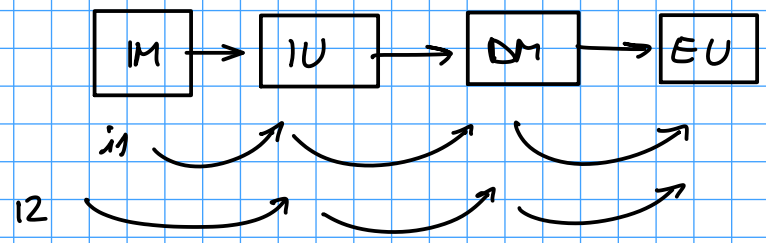
passaggio di parametri x riferimento

Interleaving





```
        CLEAR  Rsum
loop:   LOAD   Rx, Ri, R1
        ADD    Rsum, R1, Rsum
        INC    Ri
        IF2    Ri, RN, loop
out:    DIV    Rsum, RN, Rang
        END
```



CLEAR
LOAD
ADD
INC
IF2
DIV

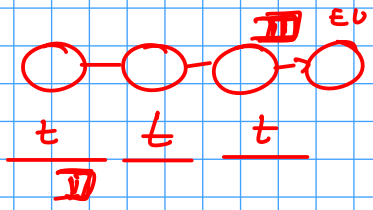
ADD R0 R0 Rsum

→ IM IU EU
→ IM IU DM EU
→ IM IU EU
→ IM IU EU
→ IM IU
→ IM IU EU

5/6

A	B
C S	D ST
E '	F S
G	H

→ 2 EU
→ 0 EU
→ 1 EU



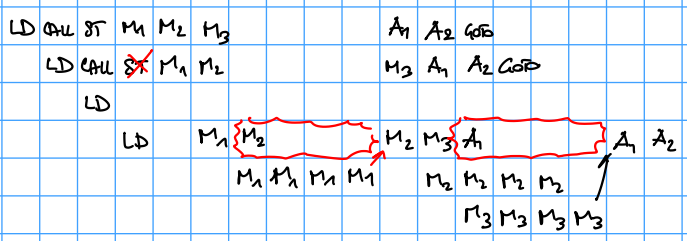
Simulazione corretta del codice con passaggio di parametri x registro (codice in rosso nella prima pagina)

```
loop: LOAD Rx, Ri, Rq
      CALL Rf, Rret
      STORE Rx, Ri, R10
```

```
f: MUL Rq Rq R2
   MUL R2 Ra R3
   MUL Rq Rb, R4
   ADD R3 R4 R5
   ADD R5, Rc, R10
   GOTO Rret
   xx
```

dipendenza EU EU !

passaggio di parametri x registro



avrei potuto ottimizzare il codice (poco)

```
MUL Rq Rq R2
MUL Rq Rb R4
MUL R2 Ra R3
ADD
ADD
GOTO
```